Atty Docket No. JGR 1008-1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

Muljadi SULISTIO et al.

Application No.: 10/026,366

Confirmation No.: 8061

Filed: 18 December 2001

Title: **METHOD AND APPARATUS FOR DECLARATIVE ERROR HANDLING AND PRESENTATION**

Group Art Unit: 2178

Examiner: Kyle R. STORK

CUSTOMER NO. 22470

MAIL STOP APPEAL BRIEF – PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

TRANSMITTAL OF APPEAL BRIEF

Sir:

In connection with the above-referenced U.S. patent application, enclosed herewith are the following papers:

[X]   *Appeal Brief*, and
[X]   *Check No. 3438* in the amount of $500.00 in payment of the filing fee associated with this document.

The Commissioner is hereby authorized to charge any underpayment of fees, or credit any overpayment, to our Deposit Account No. 50-0869 (Attorney Docket No. JGR 1008-1).

Respectfully submitted,

Dated: 3 October 2005

Ernest J. Beffel, Jr.
Registration No. 43,489

HAYNES BEFFEL & WOLFELD LLP
P.O. Box 366
Half Moon Bay, CA 94019
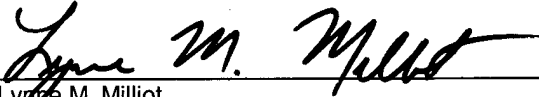Telephone: (650) 712-0340
Facsimile: (650) 712-0263

Page 1 of 1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Group Art Unit: 2178
Muljadi Sulistio et al.	
Application No.: 10/026,366	Examiner: Kyle R. Stork
Confirmation No.: 8061	
Filed: 18 December 2001	CUSTOMER NO.: 22470
Title: **METHOD AND APPARATUS FOR DECLARATIVE ERROR HANDLING AND PRESENTATION**	

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

This Appeal Brief is filed in support of Appellants' appeal from the Final Office Action, mailed 05 May 2005, in this case. A Notice of Appeal was mailed on 05 August 2005.

The appropriate fee as set forth in § 41.20 (b)(2) of $500.00 is covered in the enclosed check. Should it be determined that additional fees are required, the Commissioner is hereby authorized to charge those fees to Deposit Account No. 50-0869 (Attorney Docket No. JGR 1008-1).

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest is JGR Acquisitions, Inc., the assignee of record.

II. RELATED APPEALS AND INTERFERENCES

There are no known appeals or interferences relating to this case.

III. STATUS OF CLAIMS

Claims 1-21 are pending. All have been rejected and all of the rejections are subject to this appeal.

IV. STATUS OF AMENDMENTS

The original claims are unamended. A substitute specification was filed to move program code excerpts to a CD-ROM appendix (without renumbering the paragraphs.)

V. SUMMARY OF CLAIMED SUBJECT MATTER

There are two independent claims, numbers 1 and 12, which are addressed individually. Dependent claim 3 adds to claim 1 some of the limitations of claim 12. If either claim 1 or claim 12 is allowable, claim 3 also should be allowable. Other dependent claims will rise or fall with the claims from which they depend or to which they are analogous.

For ease of review, the reader may prefer to work from the original specification, which has paragraph numbering consistent with the substitute specification, as the original includes code samples in context, instead of in separate CD-ROM files.

Claim 1 presents a computer-implemented method for error processing and reporting during validation of a business document in a client-server environment, as described in [0054] with reference to FIG. 12 and in [0063]-[0064] with reference to FIG. 20. The method includes two phases. First, accessing a first self-describing, structured document ([0054]; [0063]; FIG. 20, ref 2001) having a document type and validating the first document against a schema (ref 2011) corresponding to the document type. For any errors detected ([0064]; ref 2004), the method includes generating a second self-describing, structured document (ref 2012) that includes at least one error identifier and a path specification identifying a node within the primary document corresponding to the detected error. Second, the two documents are

combined (ref 2005), applying a declarative transformation to produce user interface character string (FIG. 12, ref 1213). This error report includes path specifications for nodes in the first document, values for nodes in the first document and at least one error message corresponding to the at least one error identifier. The error report is transmitted.

Claim 12 includes many of the same elements as claim 1, substituting for validating the first document against a schema, instead validating the first document against a set of business processing rules applicable to the document type and an intended recipient of the first document, again as described in [0054] and [0063]. Therefore, it presents a computer-implemented method for error processing and reporting during validation of a business document in a client-server environment. The method includes two phases. First, accessing a first self-describing, structured document having a document type and validating the first document against a set of business processing rules applicable to the document type and an intended recipient of the first document. For any errors detected, the method includes generating a second self-describing, structured document that includes at least one error identifier and a path specification identifying a node within the primary document corresponding to the detected error. Second, the two documents are combined, applying a declarative transformation to produce user interface character string. This error report includes path specifications for nodes in the first document, values for nodes in the first document and at least one error message corresponding to the at least one error identifier. The error report is transmitted.

Claim 3, which depends from claim 1, adds the element (parallel to claim 12) of validating the first document against a set of business processing rules to generate a third structured document. Accordingly, claim 3 should be allowable if either claim 1 or claim 12 is allowable.

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

The claims were rejected under 35 U.S.C. § 102(b) as anticipated by Ogbuji, "Validating XML with Schematron" (2000). Ogbuji is an eight page web-published introduction to Schematron. In contrast, this application proposes to improve on Schematron.

First, whether it was improper to reject claim 1 under 35 U.S.C. § 102(b) as anticipated by Ogbuji?

Second, whether it was improper to reject claim 12 under 35 U.S.C. § 102(b) as anticipated by Ogbuji?

Third, whether it was improper to reject claim 3 under 35 U.S.C. § 102(b) as anticipated by Ogbuji?

VII. ARGUMENT

A.   Rejection of Independent Claim 1 and Dependent Claims 2-11 Under Section 102(b) as Anticipated by Ogbuji was Improper

Rejection of independent claim 1 and dependent claims 2-11 under section 102(b) was improper for the reasons set forth below.

Claim 1 includes the limitations:

validating the first document against a schema corresponding to the document type;

generating a second self-describing, structured document including, for any detected errors,

> *at least one error identifier; and*

> *a path specification identifying a node within the primary document corresponding to the detected error;*

applying a declarative transformation to the first and second documents, producing a user interface character string, including a plurality of

> *path specifications for nodes in the first document; and*

> *values for nodes in the first document; and*

> *at least one error message corresponding to the at least one error identifier;*

These limitations are not found in Ogbuji.

The Examiner's rejections are very succinct, consisting of a restatement of the claim followed by citations to Ogbuji and the rationale, "The examiner has sited [sic] sections within the Obuji reference to address the claimed limitations." FOA, p. 6 (near bottom). As Appellants see the rejection, the Examiner is relying on two different single pass examples of using native XSLT (page 4, code sample 2) and using Schematron (page 6, paragraph 3) as alternative bases for rejecting the inventive two-pass method. We traverse the examples individually.

1. Rejection Based on Page 4, Code Sample 2 was Improper

The Examiner's rejection based on native XSLT (page 4, code sample 2) should be reversed because the single pass example of generating the string "Invalid XML" lacks the claimed elements. The code sample and prefatory text are reproduced below:

Before digging into Schematron, I'll demonstrate how XSLT can easily be used to validate XML instances. Let's go back our previous example.

```
<shortStory author='AUTHOR1'>
   <character name='CHARACTER1'/>
   <character name='CHARACTER2'>
</shortStory>

<anthology author='AUTHOR1'>
    <shortStory>
        <character name='CHARACTER1'/>
        <character name='CHARACTER2'>
    </shortStory>
</anthology>
```

A template can be created that returns "Invalid XML" if a *shortStory* element ha an *author* attribute when it's contained in an *anthology* element.

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    <xsl:template match='shortStory'>
        <xsl:if test='../anthology and @author'>
            Invalid XML
        </xsl:if>
    </xsl:template>
</xsl:stylesheet>
```

One can see from this reproduction that (1) the error condition being tested is whether "a *shortStory* element has an *author* attribute when it's contained in an *anthology* element"; (2) only one pass of error processing is described; (3) only the string "Invalid XML" is returned if an error is detected.

This XSLT ground of rejection is improper because code sample 2 does not include any of the claimed features. (1) It is a single pass approach that does not include a two-pass (potentially pipelined) process of creating an errors document (second document) and merging it with the source document (first document) to produce an error report (user interface character string). (2) It does not include

validating a first document against a schema – it applies a business processing rule. (3) It does not include generating a second self-describing, structured document for any detected errors – it just reports "Invalid XML". And, we could list every element of the claim and show that none of them are met.

Therefore, it was improper to reject claim 1 under § 102(b) based on the XSLT example, Ogbuji at page 4, code sample 2.

2. Rejection Based on Page 6, Paragraph 3 was Improper

The Examiner's rejection based on Schematron language features (page 6, paragraph 3) should be reversed because the paragraph cited describes language features (*assert* and *report*) of Schematron for single pass feedback about invalid XML instances, without anticipating the claimed elements. Below, we reproduce the paragraph on which the Examiner relies plus some context:

> Rule elements may contain *assert* and *report* elements. Both elements are conditionally instantiated depending on the XPath evaluation of their *test* attribute. The only difference is that *assert* elements are instantiated if the XPath expressions evaluates to false, while the *report* elements are instantiated if it evaluates to true. (The general intent is that *assert* is used to detect errors, while *report* can be used to report affirmative qualities of an instance.)

> The assert/report mechanism is similar to the XSLT *xsl:if* element in our example stylesheet above, which also has a *test* attribute that determines if the contents of the *xsl:if* element are instantiated in the resulting XML tree.

> Note that a node can only be the context of a single *rule* (the first matching rule the processor comes across) within a *pattern*. However, a node can be matched multiple times within different patterns. Thus pattern groupings are important. Every match of a context node can be considered a discrete constraint.

> These elements allow authors of Schematron schemas to provide functional (and humanly readable) feedback about invalid XML instances. The user-defined feedback makes Schematron's unique approach to schema declaration more powerful than other schema languages.

> Finally, *assert* and *report* elements have a *name* element to use for substituting the name of an element into the output stream. The *name* element has an optional *path* attribute which returns the node whose tag name will be inserted in place of the *name* element. If the *path* attribute isn't specified the name of the current context node is used instead. This element is often used by *assert* and *report* elements to identify the tag name of an offending element within the validation message.

The last paragraph is the one the Examiner cited.

This basis for rejection is closer than the code sample 2 basis, because it says that Schematron supports providing the user something beyond the message "Invalid XML". Still, the existence of Schematron language elements that support providing "functional (and humanly readable) feedback" is not enough to sustain a § 102(b) rejection.

The cited reference to *assert* and *report* language elements of Schematron does not anticipate claim 1. (1) The use suggested is a single pass approach that does not include a two-pass process of creating an errors document (second document) and transforming both it and the source document (first document) to produce an error report (user interface character string). When the claim is considered as a whole, the reference is not an anticipation. (2) The reference does not include validating a first document against a schema, as distinct from a business processing rule, because so-called Schematron schemas are validation templates that supplement, not replace DTDs. One of skill in the art will understand the DTDs to be schemas and the Schematron templates to be business processing rules. This is supported by the specification and by differentiation between claims 1-2 and 12-13. In dependent claim 13, Schematron is given as an example of a business processing rule; in claim 2, SOX is an example of a schema. (3) The discussion of these language elements does not include any suggestion to generate a second self-describing, structured document for any detected errors. It leaves to the reader's imagination how to use *assert* and *report*. (4) There is no mention of applying a declarative transformation to the first and second documents in a second pass. Taken as a whole, Schematron ranks as a dependent claim (claim 13) that expands on a single element of claim 1, not as an anticipation of every element of the claimed method. This should not be surprising, given that the application references and improves upon Schematron.

Having expanded the Examiner's shorthand citations into two grounds of rejection and having shown that neither ground of rejection is sustainable on appeal, rejection of claim 1 should be reversed.

Claims 2-11 that depend from claim 1 should be allowable for at least the same reasons as claim 1.

B.   Rejection of Independent Claim 12 and Dependent Claims 13-21 Under Section 102(b) as Anticipated by Ogbuji was Improper

Claim 12 can be addressed briefly. It parallels claim 1, substituting validation against business processing rules for validation against a schema. The substituted element reads:

> *validating the first document against a set of business processing rules applicable to the document type and an intended recipient of the first document;*

The elements of claim 12 are not found in Ogbuji.

Most of the analysis above applies to claim 12, as the Examiner applies both the XSLT and Schematron examples in the same words to both claims 1 and 12. *See,* FOA at page 6. All of the reasons above that rejection of claim 1 was improper apply to claim 12, as well, except regarding validation against a schema.

Claim 12 requires validating against business processing rules, which both the XSLT and Schematron examples touch upon. However, the substituted element specifies *"business processing rules applicable to the document type and an intended recipient".*

The XSLT example at page 4, code sample 2, does not test the document type before applying the template and ignores any recipient. The code sample applies to any document type, depending only on the presence of an XML element named "shortStory". *See,* code sample 2, line 3 ("match='shortStory'"). The XML document processed in the XSLT example does not have a recipient, as it verifies a bibliographic record for a *shortStory.* Rejection of claim 12 based on the XSLT example was, therefore, improper for this additional reason, that it does not teach the substituted claim element.

The Schematron language features passage (page 6, paragraph 3) does not address the limitation *"business processing rules applicable to the document type and an intended recipient",* as one can see by reviewing the text reproduced above. In the

Page 7

claimed process, Schematron is an example (claim 13) of a business processing rule engine that can be applied as claimed, but the reference does not teach sets of Schematron rules qualified and applied based on an XML document type and an intended recipient. Rejection of claim 12 based on the Schematron example was, therefore, improper for this additional reason.

Having considered the limitations added to claim 12 and having shown that neither of the Examiner's grounds of rejection is sustainable on appeal, rejection of claim 12 should be reversed.

Claims 13-21 that depend from claim 12 should be allowable for at least the same reasons as claim 12.

C.   Rejection of Dependent Claim 3 as Anticipated by Ogbuji was Improper

Claim 3 adds to claim 1 part of the validation by business processing rules limitation of claim 12.

> *The method of claim 2, further including validating the first document against a set of business processing rules and generating a third self-describing, structured document, wherein the declarative transformation is further applied to the third document.*

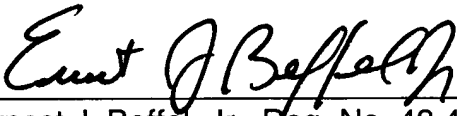If either claim 1 or 12 is allowable, claim 3 should be as well.

CONCLUSION

In view of the foregoing, Applicant/Appellant asks that this honorable Board reverse the Examiner's rejections of the claims. In addition, it is submitted that all claims are now allowable, and a notice of intent to issue a patent is respectfully requested.

The Commissioner is hereby authorized to charge any fee determined to be due in connection with this communication, or credit any overpayment, to our Deposit Account No. 50-0869 (File No. JGR 1008-1).

Respectfully submitted,

Dated: 3 October 2005

Ernest J. Beffel, Jr., Reg. No. 43,489
Attorney for Patent Owner

HAYNES BEFFEL & WOLFELD LLP

P.O. Box 366
751 Kelly Street
Half Moon Bay, CA 94019
Telephone: 650.712.0340
Facsimile: 650.712.0263

CLAIMS APPENDIX

1. (Original) A method for error processing and reporting during validation of a business document in a client-server environment, the method including:

accessing a first self-describing, structured document having a document type;

validating the first document against a schema corresponding to the document type;

generating a second self-describing, structured document including, for any detected errors,

at least one error identifier; and

a path specification identifying a node within the primary document corresponding to the detected error;

applying a declarative transformation to the first and second documents, producing a user interface character string, including a plurality of

path specifications for nodes in the first document; and

values for nodes in the first document; and

at least one error message corresponding to the at least one error identifier; and

transmitting the user interface character string.

2. (Original) The method of claim 1, wherein the schema is compliant with any version of a SOX standard.

3. (Original) The method of claim 2, further including validating the first document against a set of business processing rules and generating a third self-describing, structured document, wherein the declarative transformation is further applied to the third document.

4. (Original) The method of claim 1, wherein the declarative transformation is compliant with an XSLT standard.

5. (Original) The method of claim 3, wherein the declarative transformation is compliant with an XSLT standard.

6. (Original) The method of claim 1, wherein the user interface character string is compliant with an HTML standard.

7. (Original) The method of claim 3, wherein the user interface character string is compliant with an HTML standard.

8. (Original) The method of claim 5, wherein the user interface character string is compliant with an HTML standard.

9. (Original) The method of claim 1, wherein the user interface character string is compliant with an XML standard.

10. (Original) The method of claim 3, wherein the user interface character string is compliant with an XML standard.

11. (Original) The method of claim 5, wherein the user interface character string is compliant with an XML standard.

12. (Original) A method for error processing and reporting during validation of a business document in a client-server environment, the method including:

accessing a first self-describing, structured document having a document type;

validating the first document against a set of business processing rules applicable to the document type and an intended recipient of the first document;

generating a second self-describing, structured document including, for any detected errors,

at least one error identifier; and

a path specification identifying a node within the primary document corresponding to the detected error;

applying a declarative transformation to the first and second documents, producing a user interface character string, including a plurality of

path specifications for nodes in the first document; and

values for nodes in the first document; and

at least one error message corresponding to the at least one error identifier; and

transmitting the user interface character string.

13. (Original) The method of claim 12, wherein the business processing rules are Schematron-compliant.

14. (Original) The method of claim 12, wherein the declarative transformation is compliant with an XSLT standard.

15. (Original) The method of claim 13, wherein the declarative transformation is compliant with an XSLT standard.

16. (Original) The method of claim 12, wherein the user interface character string is compliant with an HTML standard.

17. (Original) The method of claim 13, wherein the user interface character string is compliant with an HTML standard.

18. (Original) The method of claim 15, wherein the user interface character string is compliant with an HTML standard.

19. (Original) The method of claim 12, wherein the user interface character string is compliant with an XML standard.

20. (Original) The method of claim 13, wherein the user interface character string is compliant with an XML standard.

21. (Original) The method of claim 15, wherein the user interface character string is compliant with an XML standard.